

0110745 US 00

JC872 U.S. PTO  
09/863499  
05/23/01

日本国特許庁  
PATENT OFFICE  
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されて  
いる事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed  
with this Office.

出願年月日  
Date of Application:

2000年 5月25日

出願番号  
Application Number:

特願2000-155010

願人  
Applicant(s):

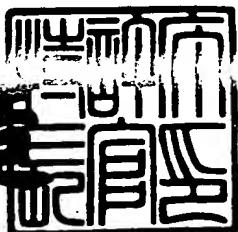
ソニー株式会社

CERTIFIED COPY OF  
PRIORITY DOCUMENT

2001年 3月 9日

特許庁長官  
Commissioner,  
Patent Office

新川謹造



【書類名】 特許願  
【整理番号】 0000483503  
【提出日】 平成12年 5月25日  
【あて先】 特許庁長官 近藤 隆彦 殿  
【国際特許分類】 G06F 9/00  
【発明者】  
【住所又は居所】 東京都品川区北品川6丁目7番35号 ソニー株式会社  
内  
【氏名】 喜多山 卓郎  
【特許出願人】  
【識別番号】 000002185  
【氏名又は名称】 ソニー株式会社  
【代表者】 出井 伸之  
【代理人】  
【識別番号】 100082762  
【弁理士】  
【氏名又は名称】 杉浦 正知  
【電話番号】 03-3980-0339  
【手数料の表示】  
【予納台帳番号】 043812  
【納付金額】 21,000円  
【提出物件の目録】  
【物件名】 明細書 1  
【物件名】 図面 1  
【物件名】 要約書 1  
【包括委任状番号】 9708843  
【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 コンポーネントウェア作成方法および記録媒体

【特許請求の範囲】

【請求項1】 第1のコンポーネントウェアと、第1のコンポーネントウェアの参照元となる第2のコンポーネントウェアとのリンクを、システムにロードされたときに行うようにされたコンポーネントウェアを作成するコンポーネントウェア作成方法において、

外部から参照されるシンボル名を該シンボル名を識別するシンボルIDに置き換えて第1のコンポーネントウェアを生成する参照先コンポーネントウェア生成のステップと、

上記第1のコンポーネントウェアに対して参照元となる第2のコンポーネントウェア中の外部に参照するシンボル名を、該シンボル名を識別するシンボルIDに置き換えて上記第2のコンポーネントウェアを生成する参照元コンポーネントウェア生成のステップと

を有し、

上記シンボルIDに基づき上記第1および第2のコンポーネントウェアをリンクするようにしたことを特徴とするコンポーネントウェア作成方法。

【請求項2】 請求項1に記載のコンポーネントウェア作成方法において、

上記第1のコンポーネントウェアには、該第1のコンポーネントウェアを識別するためのコンポーネントIDが付され、上記第2のコンポーネントウェアは、上記コンポーネントIDと上記シンボルIDとにに基づき上記第1のコンポーネントウェアを参照し、上記リンクを行うようにしたことを特徴とするコンポーネントウェア作成方法。

【請求項3】 請求項1に記載のコンポーネントウェア作成方法において、

上記参照先コンポーネントウェア生成のステップで上記第1のコンポーネントウェアを上記生成する際に、上記シンボル名と上記シンボルIDとの対応関係を示すリストを作成するリスト作成のステップをさらに有し、

上記参照元コンポーネントウェア生成のステップで上記第2のコンポーネントウェアを上記生成する際は、上記リスト作成のステップで作成された上記リスト

を用いて、上記シンボル名を上記シンボルIDに置き換えるようにしたことを特徴とするコンポーネントウェア作成方法。

【請求項4】 請求項3に記載のコンポーネントウェア作成方法において、上記第1のコンポーネントウェアを再び生成する際に、上記シンボル名を削除した場合には、上記削除された上記シンボル名に対応する上記シンボルIDが上記削除された旨を上記リストに記録するようにし、新たなシンボルを追加する場合には、上記リストに基づき該新たなシンボルに対し、新たなシンボルIDを付するようにしたことを特徴とするコンポーネントウェア作成方法。

【請求項5】 システムにロードされた際に外部のコンポーネントウェアから参照されリンクされるコンポーネントウェアが記録された記録媒体において、外部から参照されるシンボル名が該シンボル名を識別するシンボルIDに置き換えられ、上記シンボルIDが該シンボルIDと対応するシンボルの参照場所情報と関連付けられて記述されると共に、コンポーネントウェア自身を識別するコンポーネントIDが記述されたシンボル情報テーブルを有するコンポーネントウェアが記録されたことを特徴とする記録媒体。

【請求項6】 システムにロードされた外部のコンポーネントウェアを参照してリンクするコンポーネントウェアが記録された記録媒体において、

参考先のコンポーネントウェアに付された識別情報であるコンポーネントIDと、上記参考先のコンポーネントウェア内で参照されるシンボルの識別情報であるシンボルIDとが対とされて記述されると共に、上記コンポーネントIDで示される上記参考先のコンポーネントウェア内の上記シンボルIDで示される上記参照されるシンボルが上記参考先のコンポーネントウェア内で参照される参照場所情報が、上記コンポーネントIDと上記シンボルIDとの対に対応付けられたシンボル参照情報テーブルを有するコンポーネントウェアが記録されたことを特徴とする記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

この発明は、他のコンポーネントウェアのリンクをコンポーネントウェアのロ

ード時に動的に行うシステムにおけるリンク処理を効率的に行うことができるようとしたコンポーネントウェア作成方法および記録媒体に関する。

#### 【0002】

##### 【従来の技術】

近年では、コンピュータ装置や携帯用端末装置、デジタル放送を受信するためのセットトップボックスといった情報機器において用いられるソフトウェアは、コンポーネントウェア（以下、コンポーネントと略称する）と称される、部品として用いられるソフトウェアが組み合わされて構成される場合が多い。コンポーネントは、例えばソフトウェアプログラムが起動される際に必要に応じてロードされ、メモリ上に読み込まれることで当該ソフトウェアに組み込まれて使用される。

#### 【0003】

メモリ上に読み込まれたコンポーネントは、互いにリンクされて一つのソフトウェアを構成する。コンポーネントのリンクは、コンポーネント中に記述されているシンボル情報に基づき行われる。シンボル情報は、コンポーネントの作成時に、当該コンポーネントのソースファイル中の宣言に基づき、コンパイラによって自動的に設定される。シンボル情報は、シンボル情報を識別するためのシンボル名と、そのシンボル名に対応する相対的なアドレス情報であるシンボル値とかなる。相対的なアドレス情報は、例えば、ソースファイルがコンパイラでコンパイルされて生成されるオブジェクトファイル内のアドレスのオフセット値である。シンボル名は、例えば、十数文字の英数字および記号の組み合わせにより表される。

#### 【0004】

例えばコンポーネントは、複数の関数を含み、関数のそれぞれに対してシンボル情報が設定される。シンボル情報は、コンポーネントのソースファイルがコンパイルされたオブジェクトファイルにおいて、シンボル情報テーブルとして管理される。実行ファイルの作成時には、リンクにより、リンク元のコンポーネントに記述されたシンボル情報に基づき、リンク先のコンポーネント上のシンボル情報テーブルが参照され、コンポーネントのリンクが行われる。実行時には、リン

ク元のシンボル情報に基づきリンク先から対応する関数が呼び出される。

#### 【0005】

##### 【発明が解決しようとする課題】

プログラムのロード時にリンクを行う場合、リンク元のコンポーネントでの宣言に基づくシンボル情報がリンク先のコンポーネントに存在するかどうかを調べる必要がある。これは、上述した、リンク先のコンポーネントのシンボル情報テーブルを参照して、所望のシンボル情報の検索を行うことでなされる。

#### 【0006】

検索対象とされる外部シンボルの数が少ない場合には、短時間で処理が可能である。しかしながら、通常のライブラリなどでは、多くのシンボルを有し、検索に長い時間を要するという問題点があった。しかも、目的のシンボルの検索には、何度もシンボル名の比較を行う必要があり、長いシンボル名を解決するためには、より長い時間がかかるといった問題点があった。

#### 【0007】

また、シンボル名の比較照合を行う際には、検索対象となるシンボル名をメモリ上に展開する必要があり、大きなメモリ容量が必要となるという問題点があった。

#### 【0008】

さらに、同一のシンボル名を持つ二つ以上の異なるコンポーネントが同一のシステム上にロードされた場合、シンボル名の衝突が発生する。この場合、どのコンポーネントのシンボルが優先されるかは、ロードされた順番に依存するか、あるいは、ロードエラーとなってしまうという問題点があった。

#### 【0009】

一方、このようなシンボル名の衝突を解決するために、例えば特許番号第2655612号では、ロードされた異なるコンポーネントに同一のシンボル名が存在した場合、従来は時間的に前にロードされたコンポーネントに定義されているシンボルが優先されていたものを、新たにロードされたコンポーネントのシンボルを参照することが可能とされている。しかしながら、この特許番号第2655612号に記載の方法では、例えば互いに別の開発者によって作られた二つのコ

ンポーネントに同一のシンボル名が存在した場合でも、同一のシンボルとしてしまうため、正しい動作を期待することができないという問題点があった。

#### 【0010】

したがって、この発明の目的は、コンポーネントのリンクに要する処理時間と短縮すると共に、リンク時のメモリの使用量が少なくて済むようなコンポーネントウェア作成方法および記録媒体を提供することにある。

#### 【0011】

また、この発明の他の目的は、リンク関係にある二つ以上のコンポーネント間でのシンボル名の衝突を回避できるようなコンポーネントウェア作成方法および記録媒体を提供することにある。

#### 【0012】

##### 【課題を解決するための手段】

この発明は、上述した課題を解決するために、第1のコンポーネントウェアと、第1のコンポーネントウェアの参照元となる第2のコンポーネントウェアとのリンクを、システムにロードされたときに行うようにされたコンポーネントウェアを作成するコンポーネントウェア作成方法において、外部から参照されるシンボル名をシンボル名を識別するシンボルIDに置き換えて第1のコンポーネントウェアを生成する参照先コンポーネントウェア生成のステップと、第1のコンポーネントウェアに対して参照元となる第2のコンポーネントウェア中の外部に参照するシンボル名を、シンボル名を識別するシンボルIDに置き換えて第2のコンポーネントウェアを生成する参照元コンポーネントウェア生成のステップとを有し、シンボルIDに基づき第1および第2のコンポーネントウェアをリンクするようにしたことを特徴とするコンポーネントウェア作成方法である。

#### 【0013】

また、この発明は、システムにロードされた際に外部のコンポーネントウェアから参照されリンクされるコンポーネントウェアが記録された記録媒体において、外部から参照されるシンボル名がシンボル名を識別するシンボルIDに置き換えられ、シンボルIDがシンボルIDと対応するシンボルの参照場所情報と関連付けられて記述されると共に、コンポーネントウェア自身を識別するコンポーネ

ントIDが記述されたシンボル情報テーブルを有するコンポーネントウェアが記録されたことを特徴とする記録媒体である。

【0014】

また、この発明は、システムにロードされた外部のコンポーネントウェアを参照してリンクするコンポーネントウェアが記録された記録媒体において、参照先のコンポーネントウェアに付された識別情報であるコンポーネントIDと、参照先のコンポーネントウェア内で参照されるシンボルの識別情報であるシンボルIDとが対とされて記述されると共に、コンポーネントIDで示される参照先のコンポーネントウェア内のシンボルIDで示される参照されるシンボルが参照先のコンポーネントウェア内で参照される参照場所情報が、コンポーネントIDとシンボルIDとの対に対応付けられたシンボル参照情報テーブルを有するコンポーネントウェアが記録されたことを特徴とする記録媒体である。

【0015】

上述したように、請求項1に記載の発明は、参照先である第1のコンポーネントウェアは、外部から参照されるシンボル名をシンボル名を識別するシンボルIDに置き換えて生成され、参照元である第2のコンポーネントウェアは、外部に参照するシンボル名を、シンボル名を識別するシンボルIDに置き換えて生成され、シンボルIDに基づき第1および第2のコンポーネントウェアをリンクするようしているため、シンボルIDを短く設定すれば、シンボル名の照合の時間が短縮され第1および第2のコンポーネントウェアのリンク処理を高速化することができると共に、シンボル名の照合に必要なメモリの使用量を少なく抑えることができる。

【0016】

また、請求項5に記載の発明によれば、外部から参照されるシンボル名がシンボル名を識別するシンボルIDに置き換えられ、シンボルIDがシンボルIDと対応するシンボルの参照場所情報と関連付けられて記述されると共に、コンポーネントウェア自身を識別するコンポーネントIDが記述されたシンボル情報テーブルを有するコンポーネントウェアが記録媒体に記録されるため、記録媒体から読み出されたコンポーネントウェアは、他のコンポーネントウェアからシンボル

IDで以てシンボルが参照されたときに、シンボル情報テーブルに基づきシンボルの参照場所を知ることができる。

#### 【0017】

また、請求項6に記載の発明によれば、参照先のコンポーネントウェアに付された識別情報であるコンポーネントIDと、参照先のコンポーネントウェア内で参照されるシンボルの識別情報であるシンボルIDとが対とされて記述されると共に、コンポーネントIDで示される参照先のコンポーネントウェア内のシンボルIDで示される参照されるシンボルが参照先のコンポーネントウェア内で参照される参照場所情報が、コンポーネントIDとシンボルIDとの対に対応付けられたシンボル参照情報テーブルを有するコンポーネントウェアが記録媒体に記録されるため、記録媒体から読み出されたコンポーネントウェアは、他のコンポーネントウェアを参照するときに、シンボル参照情報テーブルに基づき他のコンポーネントウェア内の参照するシンボルの参照場所を知ることができる。

#### 【0018】

##### 【発明の実施の形態】

以下、この発明の実施の一形態について、図面を参照しながら説明する。図1は、この発明の実施の一形態に適用可能なシステムの概念を示す。例えば携帯用情報機器やコンピュータ装置といった情報機器上に構成されるターゲットシステム1において、所定のソフトウェアの動作を実現するために、互いに関数などの呼び出しを行う関係にある複数のコンポーネント4がロードされる。

#### 【0019】

コンポーネントは、他のプログラムの部品として用いることができるプログラムを指す。コンポーネントは、例えば複数の関数を有し、他のコンポーネントとリンクされることで、他のコンポーネントやプログラムから関数の呼び出しや変数の参照などがなされる。以下では、説明のため、コンポーネント4Aの記述に基づき、コンポーネント4B内に定義されている関数や変数の参照が行われるものとする。すなわち、コンポーネント4Aがリンク元、コンポーネント4Bがリンク先とされ、コンポーネント4Aに対してコンポーネント4Bがリンクされる。図1の例では、リンク元のコンポーネント4A中の記述「func()」により、リ

ンク先である他のコンポーネント4Bが参照され、コンポーネント4Bが有する関数func()が呼び出される。

#### 【0020】

ターゲットシステム1にロードされていないコンポーネント4は、例えばターゲットシステム1に接続された2次記憶装置3からロードすることができる（コンポーネント4C）。2次記憶装置3としては、ハードディスク装置やCD-R ROM(Compact Disc-Read Only Memory)、あるいは、その他の記録媒体を適用することができる。また、これに限らず、フラッシュメモリなどの半導体記憶素子を2次記憶装置3として適用させることができる。さらに、フロッピーディスクやMO(Magneto Optical Disk)などを2次記憶装置3として適用することができる。

#### 【0021】

ターゲットシステム1により、コンポーネント4Aによって指定されるコンポーネント4Bが2次記憶装置3から読み出される。読み出されたコンポーネント4Bは、メモリに展開され、ターゲットシステム1にロードされる。

#### 【0022】

これに限らず、コンポーネント4は、ターゲットシステム1に接続されたネットワーク2を通して、サーバシステム50からダウンロードして、ターゲットシステム1にロードすることもできる（コンポーネント4D）。

#### 【0023】

図2は、ターゲットシステム1が構成される情報機器の一例の構成を示す。ここでは、ターゲットシステム1が携帯用情報機器10上に構成されるものとして説明する。勿論、ターゲットシステム1は、これに限らず、パーソナルコンピュータやセットトップボックス、携帯電話装置、家庭用ゲーム装置など、他の情報機器に構成されるようにしてもよい。バス11に対してCPU(Central Processing Unit)12、RAM(Random Access Memory)13、ROM(Read Only Memory)13および上述の2次記憶装置3が接続される。RAM13は、CPU12のワークメモリである。ROM14には、この携帯用情報機器10を動作させる基本的なソフトウェアであるOS(Operating System)や、コンポーネントをロード

するためのコンポーネントローダ、コンポーネント同士のリンクを行うリンク、起動時に必要とされるプログラムといった、その他のプログラムが予め記憶される。

#### 【0024】

2次記憶装置3としては、例えば不揮発性で且つ書き替えが可能なメモリであるフラッシュメモリを用いることができる。この場合、2次記憶装置3は、換装可能とすることができます。2次記憶装置3は、これに限らず、ハードディスク装置を用いることができる。また、所定のドライブ装置を用いてMOやフロッピーディスクなどの記録媒体を2次記憶装置3として用いることができる。

#### 【0025】

バス11に対して、さらに、表示部16、入力部17および通信インターフェイス18とが接続される。表示部16は、例えばLCD(Liquid Crystal Display)からなる。CPU12から表示制御コマンドが発行され、この表示制御コマンドに基づき所定の表示が表示部16になされる。入力部17は、例えば表示部16上に設けられ、表示部16の表示が透過可能なタッチパネルからなる。通信インターフェイス18は、無線あるいは有線通信によるネットワーク2と接続され、この携帯用情報機器10と外部との通信を制御する。

#### 【0026】

携帯用情報機器10が起動されると、CPU12によりROM14に格納されたOSなどのプログラムが起動され、ターゲットシステム1が構成される。例えば入力部17からなされた指示により、ターゲットシステム1上で所定のプログラムが起動され、プログラム中の記述に従い、必要なコンポーネントが2次記憶装置3からRAM13に読み込まれ、ロードされる。

#### 【0027】

コンポーネントにおけるリンクの指定は、コンポーネントの作成時になされる。リンクは、リンク元とリンク先とで対応するシンボルを参照することでなされる。この発明では、リンク先となるコンポーネントは、当該コンポーネントを示すコンポーネントIDを付されると共に、当該コンポーネントに含まれるシンボル情報が、例えば通し番号からなるシンボルIDで置き換えられた、シンボル情

報テーブルを有する。

#### 【0028】

一方、リンク元のコンポーネントは、リンク先となるコンポーネントのコンポーネントIDと、そのコンポーネントIDで示されるコンポーネント中の参照するシンボルのシンボルIDとが対とされる。そして、このコンポーネントIDとシンボルIDとの対と、シンボルIDで示されるシンボルの、リンク先コンポーネント中の相対的なアドレスとが対応付けられた、シンボル参照情報テーブルを有する。

#### 【0029】

リンク元のコンポーネントは、シンボル参照情報テーブルに基づき、リンク先のコンポーネントおよびそのコンポーネント中で参照されるシンボルの相対的な位置を知ることができる。シンボルIDは、一般的に、オリジナルのシンボル名よりも短く設定することができる。このため、オリジナルのシンボル名がより短いシンボルIDで置き換えられるので、リンク先の検索が高速に行えると共に、メモリの使用量が抑えられる。

#### 【0030】

図3～図5を用いて、この実施の一形態におけるコンポーネントの生成方法について説明する。なお、以下に説明するコンポーネントの生成は、ターゲットシステム1とは異なるシステムで実行することができる。

#### 【0031】

図3は、リンク先となるコンポーネント（参照先コンポーネント4B）の一例の生成方法を示す。リンク先となるコンポーネントは、外部から参照されるシンボルを持つ。図3Aは、ソースファイル10において宣言された関数に対してシンボルが宣言された様子を示している。このソースファイルをコンパイラによってコンパイルすることで、図3Bに示されるオブジェクトファイル11が生成される。オブジェクトファイルは、ソースファイルから実行型のファイルを作成する途中で生成される中間ファイルである。

#### 【0032】

ソースファイル10中で宣言された関数や変数で、他のコンポーネントから参

照されるシンボルは、オブジェクトファイル11中のシンボル情報テーブル12に列挙される。このとき、ソースファイル10においてなされたシンボルの宣言は、コンパイラによってシンボル名に置き換えられる。ここでは、シンボル情報テーブル12に書かれているシンボル名は、ソースコード10における記述に基づき生成された、オリジナルのシンボル名が用いられる。

## 【0033】

図3の例では、シンボル名は、「\_func1\_6Class1P6Class1」というような、十数文字の英数字および記号の組み合わせからなる文字列になっている。シンボル名には、宣言されたシンボルのオブジェクトファイル11における相対的なアドレス（オフセット値）を示すシンボル値が付される。これら、シンボルの宣言のシンボル名への置き換えとシンボル値の付加は、コンパイラによって自動的になされる処理である。

## 【0034】

このオブジェクトファイル11から、リンク先となる参照先コンポーネント4Bが作成される。このとき、この発明の実施の一形態では、オブジェクトファイル11中のシンボル情報テーブル12において、図3Cに一例が示されるように、シンボル名をシンボルIDに置き換えた、シンボル情報テーブル14が作成される。シンボルIDは、当該コンポーネント中で他のシンボルIDと重複しないような、任意の文字列とされる。図3Cの例では、上述のシンボル名「\_func1\_6Class1P6Class1」に対して「S3」がシンボルIDとして設定される。シンボル名がシンボルIDに置き換えられたシンボル情報テーブル14には、さらに、当該コンポーネント4Bを示すコンポーネントID（この例では、「C45」）が設定される。コンポーネントIDは、コンポーネント毎に一意に割り当てられる。

## 【0035】

また、この実施の一形態においては、コンポーネントファイル生成時に、作成されるコンポーネントファイルとは別に、外部シンボル情報ファイル13が作成される。外部シンボル情報ファイル13は、図4に一例が示されるように、コンポーネントIDに示されるコンポーネントについて、オリジナルのシンボル名と

シンボルIDとの対応関係が各エントリ（行）に列挙される。

#### 【0036】

さらに、コンポーネント4Bがバージョンアップなどにより再作成されシンボルが削除された場合、図3Cに「DELETE」の行で示されるように、外部シンボル情報ファイル13に対して、シンボルが削除された旨が削除されたシンボルIDと共に記述される。このコンポーネントに対して新たにシンボルが追加される場合には、削除されたシンボルを含めて、過去に使われていないシンボルIDが使用される。このようにすることで、過去に使用されたシンボルIDが再使用されるのが防がれ、同一名で互いにバージョンの異なる複数のコンポーネントにおいて、シンボルIDの唯一性が保証される。

#### 【0037】

図5は、リンク元となるコンポーネント（参照元コンポーネント4A）の一例の生成方法を示す。リンク元となるコンポーネント4Aは、外部のコンポーネントにおいて定義されている関数の呼び出しや、変数の参照を行う。図5Aは、ソースファイル20において他のコンポーネントの関数が参照されていることが示されている。このソースファイルがコンパイラによってコンパイルされ、図5Bに示されるオブジェクトファイル21が生成される。

#### 【0038】

オブジェクトファイル21は、シンボル参照情報テーブル22を有する。シンボル参照情報テーブル22は、外部のコンポーネントに参照するシンボルが列挙される。このときの参照先のシンボル名は、ソースファイル20における参照を指示する記述に基づきコンパイラで生成された、オリジナルのシンボル名がそのまま用いられる。図5の例では、ソースファイル20における「func1(obj)」に対して、オブジェクトファイル21中の「\_func1\_6Class1P6Class1」がシンボル名として用いられる。シンボル名に対して、参照先のコンポーネントにおける対応する、相対的なアドレス情報が付加される。

#### 【0039】

また、シンボル参照情報テーブル22と、上述の参照先コンポーネント4Bを生成する際に作成された外部シンボル情報ファイル13とから、参照先コンポー

ネット4Bについての外部シンボル情報ファイル13'が作成される(図5B)。外部シンボル情報ファイル13と、シンボル参照テーブル22とに基づき、オリジナルのシンボル名に対応するコンポーネントIDとシンボルIDとが得られる。得られたコンポーネントIDとシンボルIDとの対と、オリジナルのシンボル名とを対応付ける。これを、シンボル参照情報テーブル22に列挙されるオリジナルのシンボル名について、全て行う。

#### 【0040】

オブジェクトファイル21と外部シンボル情報ファイル13'とから、参照元のコンポーネント4Aが生成される(図5C)。コンポーネント4Aは、参照先のコンポーネントIDおよびシンボルIDの対と、シンボルの参照アドレスとが対応付けられて列挙された、シンボル参照情報テーブル23を有する。図5Cの例では、コンポーネントIDとシンボルIDの対は、「C45\_S3」のように表現され、これがシンボルの参照場所を示すシンボル値と対応付けられる。

#### 【0041】

シンボル参照情報テーブル23は、外部シンボル情報ファイル13'を用いて、オブジェクトファイル21のシンボル参照情報テーブル22におけるオリジナルのシンボル名をコンポーネントIDとシンボルIDとの対に変換したものであり、置き換えられたシンボル名と、コンポーネント内の、シンボルを参照しているアドレス情報を有している。

#### 【0042】

すなわち、シンボル参照情報テーブル23は、オブジェクトファイル21中のシンボル参照情報テーブル22におけるオリジナルのシンボル名が、コンポーネントIDとシンボルIDの対からなる新たなシンボル名によって置き換えられたと考えることができる。

#### 【0043】

なお、ここでは、他のコンポーネントから参照されるシンボルを持つコンポーネントと、他のコンポーネントが持つシンボルを参照するコンポーネントを別々に説明しているが、これはこの例に限られない。一つのコンポーネントが他のコンポーネントから参照されるシンボルを持ち、且つ、そのコンポーネントが他の

コンポーネントが持つシンボルを参照することも可能であり、そのような場合でも、この発明を適用することができる。

#### 【0044】

次に、上述のようにして生成された参照元コンポーネント4Aと参照先コンポーネント4Bとをリンクして実行ファイルを作成する方法について説明する。図6は、ターゲットシステム1上でのコンポーネントのリンク機構を概略的に示す。ここでは、ターゲットシステム1では予め所定のプログラムが起動され、あるコンポーネント4AがRAM13上に読み込まれているものとする。このコンポーネント4Aの記述に基づき、他のコンポーネント4Bが呼び出され、コンポーネント4Aに対してリンクされる。なお、これは、実際には、OS20の制御に基づきなされる処理である。

#### 【0045】

コンポーネント4Aのシンボル参照情報テーブル23に基づき、コンポーネント4AからOS30に対して、コンポーネント4Bのリンクが要求される。このリンク要求は、OS30からリンク31に渡される。リンク31は、この要求に基づき参照元のコンポーネントのシンボル参照情報テーブル23と、参照先のコンポーネント4Bのシンボル情報テーブル14とを参照する。そして、リンク31は、これらテーブル23および14に記述されているコンポーネントIDおよびシンボルIDに基づき、参照元コンポーネント4Aと参照先コンポーネント4Bとをリンクする。

#### 【0046】

図7は、ターゲットシステム1におけるコンポーネント間のリンクを行う一例の手順を示すフローチャートである。このフローチャートによる処理は、OS30の制御に基づき、リンク31によってなされるものである。最初のステップS10で、参照元のコンポーネント4Aと参照先のコンポーネント4BとがRAM13に読み込まれ、ターゲットシステム1にロードされる。これは、例えば図示されないコンポーネントローダによって、参照元コンポーネント4Aに記述された、参照先のコンポーネント4Bのコンポーネント名やバージョン情報に基づき、2次記憶媒体3に格納されているコンポーネントから適切なコンポーネントが検

索されて、なされる。

#### 【0047】

参照元および参照先のコンポーネント4Aおよび4Bがそれぞれロードされたら、上述したように、参照元のコンポーネント4AからOS30に対して参照元コンポーネント4Bに対するリンク要求が出され、このリンク要求がOS30からリンク31に渡される。次のステップS11で、リンク31によって、参照元のコンポーネント4Aのシンボル参照情報テーブル23から、エントリが一つ取得される。ここで、シンボル参照情報テーブル23の全てのエントリに対しての処理が終了していると判断されたなら（ステップS12）、一連のフローチャートが終了される。

#### 【0048】

若し、ステップS12で、参照元情報テーブル23において処理が終了していないエントリが存在するとされたら、処理はステップS13に移行する。ステップS13では、参照元コンポーネント4Aのシンボル参照情報テーブル23に基づき、参照先コンポーネント4Bのシンボル情報テーブル14の該当するエントリが検索される。上述したように、参照元コンポーネント4Aのシンボル参照情報テーブル23におけるシンボル名は、コンポーネントIDとシンボルIDとの組み合わせで構成されているので、シンボル参照情報テーブル23の記述に基づき、参照先のコンポーネント4BのコンポーネントIDが取得される。

#### 【0049】

ステップS14では、取得されたコンポーネントIDが、リンク先である参照先コンポーネント4BのコンポーネントIDと一致しているかどうかが判断される。若し、一致していないとされれば、処理はステップS17に移行し、参照先コンポーネント4Bに対して参照元コンポーネント4Aをリンクしないままにしておく。そして、処理はステップS11に戻され、シンボル参照情報テーブル23の次のエントリが取得される。

#### 【0050】

一方、ステップS14で、ステップS13で取得されたコンポーネントIDが参照先コンポーネント4BのコンポーネントIDと一致していると判断されれば

、処理は次のステップS15に移行する。ステップS15では、リンク先となる参照先コンポーネント4Bのシンボル情報テーブル14が参照され、リンクの対象となるシンボル情報エントリが取得される。

#### 【0051】

すなわち、ステップS15では、シンボル参照情報テーブル23に基づき、ステップS13で取得されたコンポーネントIDと対応されているシンボルIDが求められる。そして、求められたシンボルIDに基づき、参照先コンポーネント4Bのシンボル情報テーブル14のエントリが検索される。

#### 【0052】

ステップS15では、上述のステップS14の検索結果に基づき、シンボルIDに対応するシンボル値が求められる。これにより、シンボル参照情報テーブル23に記述されている参照場所の参照が解決され、シンボルのリンクが行われる（ステップS16）。これら一連の処理が、参照元コンポーネント4Aのシンボル参照テーブル23の各エントリに対して行われる。

#### 【0053】

なお、参照先のコンポーネント4Bの元となるオブジェクトファイル11に用いられたオリジナルのシンボル名と、参照元コンポーネント4の元となるオブジェクトファイル21に用いられたオリジナルのシンボル名は、これらのコンポーネントの生成時には、全て、コンポーネントIDとシンボルIDとの組み合わせによるシンボル名に置き換えられている。そのため、参照先のコンポーネントの外部シンボル参照ファイル13および参照元のコンポーネントの外部シンボル参照ファイル13'、ならびに、各シンボル名を置き換える以前の、オリジナルのシンボル名に関する情報は、コンポーネント生成以降は、不要となる。

#### 【0054】

##### 【発明の効果】

以上説明したように、この発明によれば、ターゲットシステム1上で必要とされるシンボル情報は、シンボルIDに置き換えられて用いられる。シンボルIDは、オリジナルのシンボル名に比べて短くすることが可能である。これにより、シンボル名をターゲットシステム1上に格納するためのメモリの量を軽減するこ

とができると共に、コンポーネントのリンク時などにおけるシンボル検索の際の処理時間を短くすることができるという効果がある。

#### 【0055】

また、この発明によれば、参照元のシンボル参照情報にコンポーネントIDを付加している。これにより、コンポーネントのリンク時に、シンボル参照情報の参照先のシンボルがリンク先のコンポーネントに存在するかどうかを、コンポーネントIDを比較するだけで判断することができる。したがって、シンボルの検索に要する処理時間を短縮することができるという効果がある。

#### 【0056】

さらに、この発明によれば、全てのコンポーネントのシンボル名が置き換えられたものとして扱われ、また、シンボルIDに対してそれぞれのシンボルが属しているコンポーネントの識別情報（コンポーネントID）が付加されるため、異なる2つ以上のコンポーネントにおいて、同一名称のシンボル名が宣言された場合でも、これらを別々のシンボルとして扱うことができるという効果がある。

#### 【図面の簡単な説明】

##### 【図1】

この発明の実施の一形態に適用可能なシステムの概念を示す略線図である。

##### 【図2】

ターゲットシステムが構成される情報機器の一例の構成を示すブロック図である。

##### 【図3】

リンク先となるコンポーネントの一例の生成方法を示す略線図である。

##### 【図4】

外部シンボル情報ファイルの一例の構成を示す略線図である。

##### 【図5】

リンク元となるコンポーネントの一例の生成方法を示す略線図である。

##### 【図6】

ターゲットシステム上でのコンポーネントのリンク機構を概略的に示すブロック図である。

【図7】

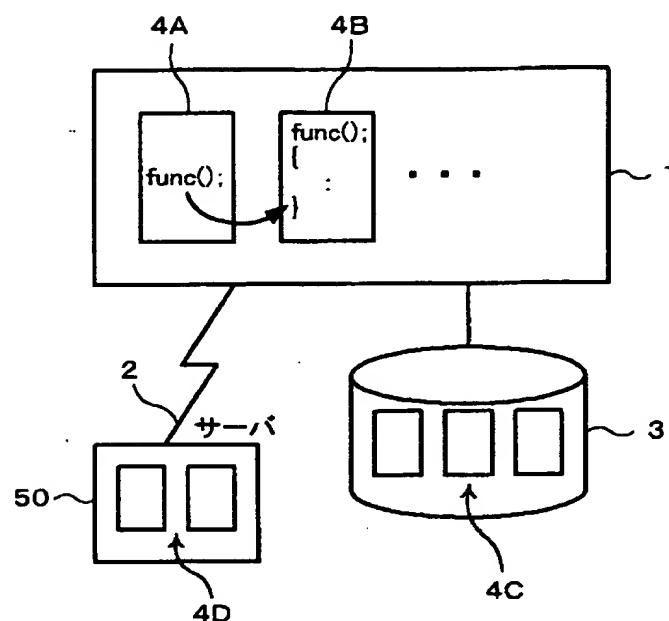
ターゲットシステムにおけるコンポーネント間のリンクを行う一例の手順を示すフローチャートである。

【符号の説明】

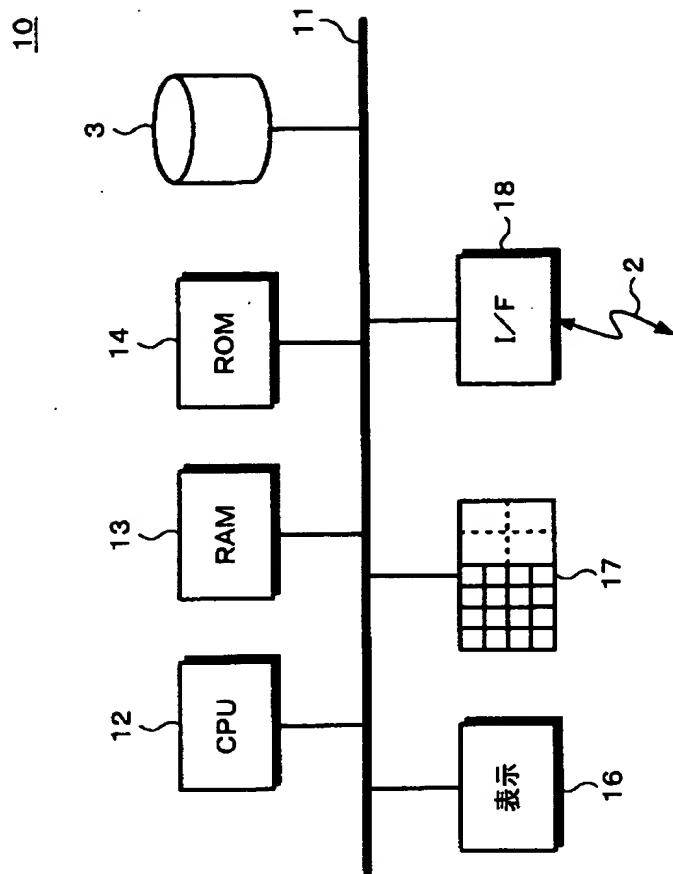
1 . . . ターゲットシステム、2 . . . ネットワーク、3 . . . 2次記憶装置、  
4 A . . . 参照元のコンポーネント、4 B . . . 参照先のコンポーネント、1 3  
, 1 3' . . . 外部シンボル参照ファイル、1 4 . . . シンボル情報テーブル、  
2 3 . . . シンボル参照情報テーブル

【書類名】 図面

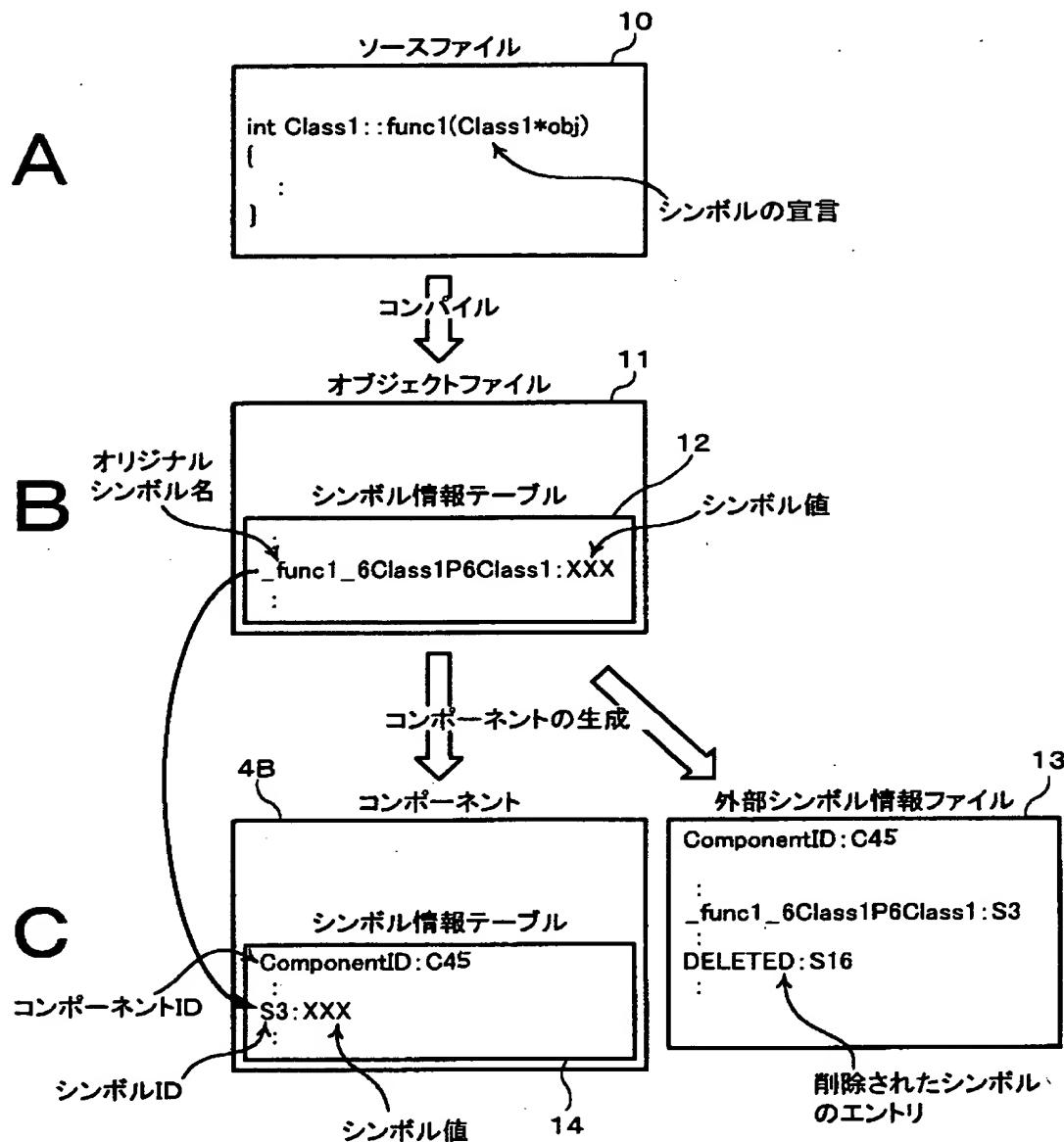
【図1】



【図2】



【図3】

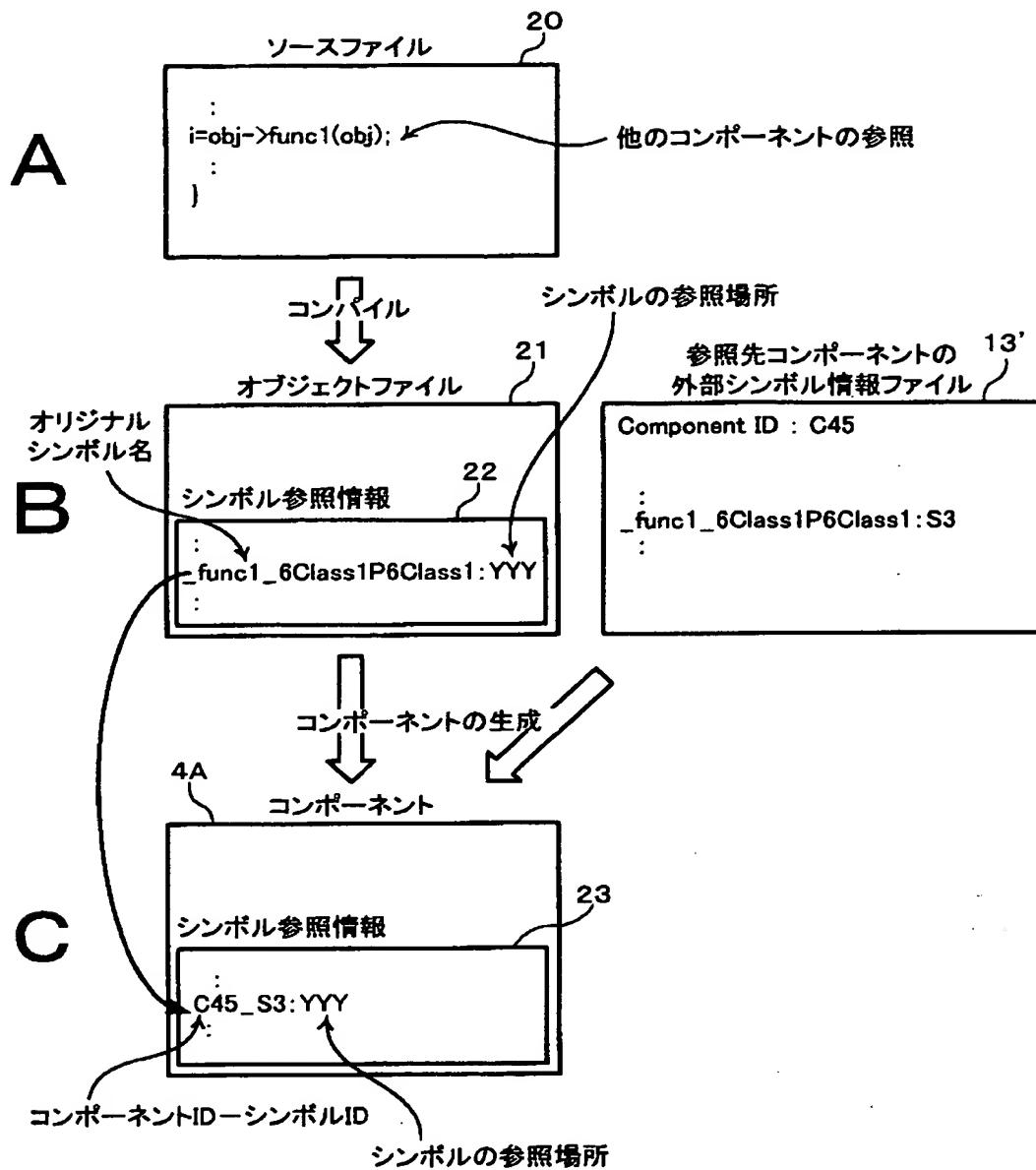


【図4】

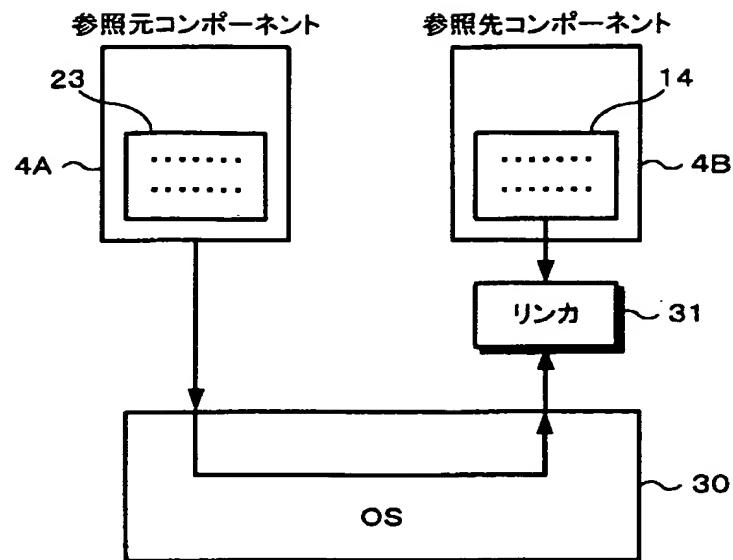
13

ComponentID	コンポーネントID
オリジナルシンボル名1	シンボルID1
オリジナルシンボル名2	シンボルID2
:	:
オリジナルシンボル名n	シンボルIDn

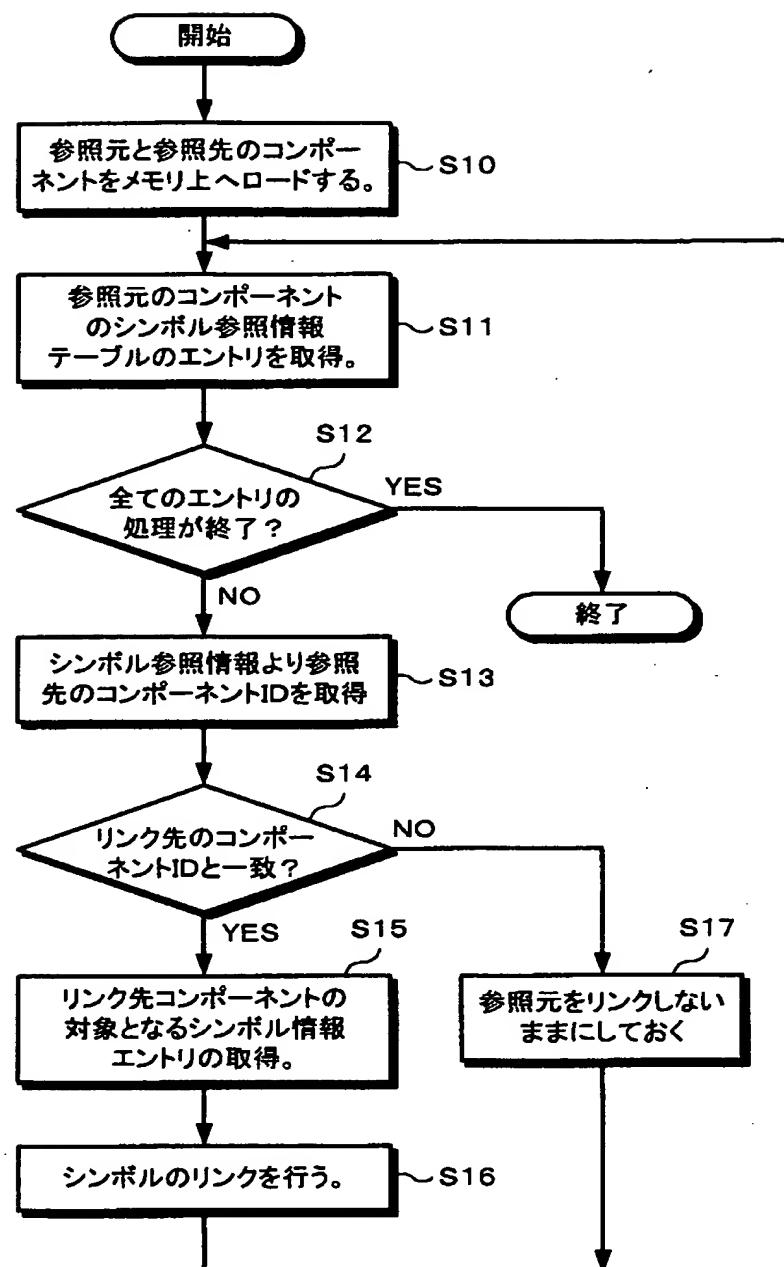
【図5】



【図6】



【図7】



【書類名】 要約書

【要約】

【課題】 コンポーネントのリンクに要する処理時間を短縮すると共に、リンク時のメモリの使用量が少なく抑える。

【解決手段】 他のコンポーネントから参照されるコンポーネントの作成に際し、コンパイル処理によりソースファイル10中のシンボルの宣言がオリジナルシンボル名に変換され、オブジェクトファイル11内のオフセットアドレスであるシンボル値と対応付けられる。参照先コンポーネント4Bの作成時に、オリジナルシンボル名がシンボルIDに置き換えられてシンボル値と対応付けられたシンボル情報テーブル14が作成される。また、オリジナルシンボル名とシンボルIDとが対応され外部シンボル情報ファイル13が作成される。参照元コンポーネントでは、ファイル13を用いて、他のコンポーネントを参照するオリジナルシンボル名がコンポーネントIDとシンボルIDとの対に置き換えられ、シンボル参照情報テーブル23が作成される。参照元および参照先コンポーネントのリンクは、テーブル23とテーブル14とに基づき、シンボルIDを用いてリンク情報の照合が行われる。シンボルIDは短いので、シンボルの照合に要する時間が短縮されると共に、メモリの使用量が少なくできる。

【選択図】 図3

出願人履歴情報

識別番号 [000002185]

1. 変更年月日 1990年 8月30日

[変更理由] 新規登録

住 所 東京都品川区北品川6丁目7番35号

氏 名 ソニー株式会社